IN THE UNITED STATES DISTRICT COURT
FOR THE NORTHERN DISTRICT OF ILLINOIS
EASTERN DIVISION

KOVE IO, INC.	§	
	§	Case No. 1:23-cv-04244
Plaintiff,	§	
	§	Hon. Mary M. Rowland
v.	§	
	§	JURY TRIAL DEMANDED
GOOGLE LLC,	§	
	§	
Defendant.	§	
	§	

DEFENDANT GOOGLE LLC'S RULE 12(B)(6) MOTION TO DISMISS KOVE IO, INC.'S FIRST AMENDED COMPLAINT

TABLE OF CONTENTS

Page

TABLE OF AUTHORITIES

Page(s)

Cases

Other Authorities

Kove IO, Inc. ("Kove") accuses Google of having directly infringed three related patents before they expired years ago. Kove alleges that two Google systems (called Colossus and Spanner) practiced four patent claims of the three Asserted Patents. The Court should dismiss Kove's First Amended Complaint (FAC) because, based on the facts pleaded, Kove has not plausibly alleged that Google infringed any Asserted Claims.

First, two of the four asserted claims require a client to request the location of data (*i.e.,* "location information") from a "location server";[1] if the location server does not have the location of the requested data, then the location server sends back a "redirect message" identifying the correct (different) location server. But, as confirmed by exhibits to the FAC, Google's accused products use a materially different, hierarchy-based approach to identify the location of data. Thus, the facts pleaded cannot plausibly support the infringement allegations.

A third asserted claim requires location servers to organize and locate data based on a "hash function." As explained more below, exhibits to the FAC show that Kove accuses a *range*-based system, not a *hash*-based system. Both range-based and hash-based systems organize data by subdividing large data tables and storing the "splits" in separate locations, but do so in critically different ways. Consider a database storing data on U.S. presidents. In a range-based system, the presidents may be ordered chronologically and split by which century they first served in office.

[1] Although the Court need not construe claims to grant this motion, certain terms have been previously construed, and none of Google's arguments conflict with those constructions. In that case, "client" was construed as "a network-attached component (which may be software or hardware) that initiates update or lookup of identifier/location mappings from a location server with location request messages," *Kove IO, Inc. v. Amazon Web Servs., Inc.*, Dkt. 484 at 21–23, 35 (N.D. Ill. 2021); "location information" in the '978 Patent was construed as "one or more identifiers and their associated locations" and in the '170 and '640 patents as "information pertaining to one or more locations of data and/or the identities of one or more location servers," *id*. at 12–19, 35; and "location server" was construed as "a network-attached component that maintains a set of identifier/location mappings that are modified or returned in response to location request messages from clients," *id*. at 19–21, 35.

The "ranges" here would be 1789–1799; 1800-1899, etc., which would organize data in four "splits" of unequal quantity. A hash-based system would instead use a hashing algorithm to equally distribute the presidents into each "split." See below a visualization of this example:

	Number of Presidents in Each Split			
Splits	Split 1	Split 2	Split 3	Split 4
Range-Based	2	23	17	4
Hash-Based	12	12	11	11

The range-based approach prioritizes ease of locating and retrieving information whereas the hash-based approach prioritizes reducing the "burden" on any given split by spreading the data more evenly but makes the information harder to locate as quickly. The FAC exhibits clarify this important difference, rendering Kove's infringement claim implausible.

The remaining asserted claim requires that the Accused Products implement a claimed "transfer protocol." Common examples of transfer protocols include FTP (file transfer protocol), HTTP (hyper-text transfer protocol), and TCP/IP (transmission control protocol and internet protocol). The claimed "transfer protocol" is none of those, and Kove's FAC provides only tautological allegations regarding a transfer protocol in the Accused Products, without alleging facts supporting an inference that the ***claimed*** transfer protocol is used. Because this does not satisfy the requirement for fact-based pleading, this remaining claim should also be dismissed.

I. SUMMARY OF THE FIRST AMENDED COMPLAINT

Kove accuses Google of having infringed U.S. Patent Nos. 7,103,640 (the "'640 Patent"), 7,233,978 (the "'978 Patent"), and 7,814,170 (the "'170 Patent") (First Am. Compl. ("FAC") ¶¶ 16–20, Dkt. 36) (collectively, the "Asserted Patents"). The Asserted Patents relate to storing and retrieving information in distributed networks. As discussed more below, the Asserted Patents disclosed alleged advancements by using "identification strings," which specify the identity of "an entity," and "location strings," which specify the location of data associated with that entity. Information is stored using a hash function and retrieved by applying the hash function to an identifier string (*e.g.,*'170 Patent, Claim 1). Or the information is retrieved by sending a requesting client either the location string or a "redirect message" that tells the client where to find the location string (*e.g.,* '170 Patent, claim 15; '640 Patent, Claim 10). Alternatively, identifiers can be stored and transferred using a "transfer protocol" (*e.g.,* '978 Patent, Claim 17).

Kove alleges direct infringement of claims 1 and 15 of the '170 Patent, (FAC ¶¶ 26–56); claim 17 of the '978 Patent, (*id.* ¶¶ 57–75); and claim 10 of the '640 Patent (*id.* ¶¶ 76–95) (collectively, the "Asserted Claims"). Kove accuses the Google products Spanner and Colossus (collectively, the "Accused Products") of having infringed the Asserted Claims. (*Id.* ¶ 22).[2]

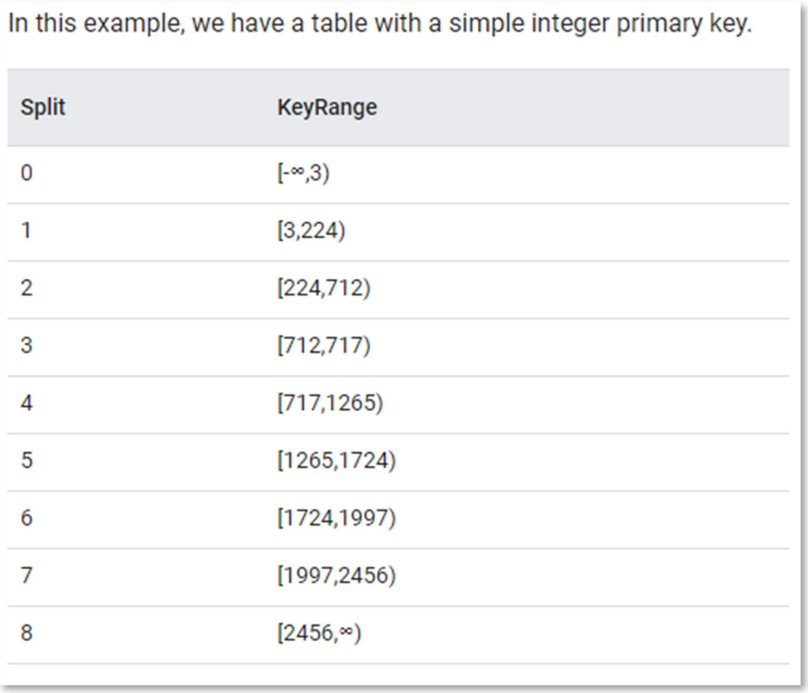
Spanner provides database functionality and Colossus is a cluster-level file system. (FAC, Ex. 8 at 2–3). At a very high level, a database service like Spanner stores "structured data," (*id.*, Ex. 10 at 1)—*e.g.*, data tables. Structured data may take the form of a table containing a column listing keys that identify particular files and additional columns listing the file's metadata like "access permissions and data location." (*Id.*, Ex. 8 at 3). Colossus, in turn, provides clients with access to stored data. (*Id.* at 3–4).

In part because the Spanner and Colossus datasets are massive, the datasets are split and stored at multiple locations. Spanner and Colossus break data into regions, and further into multiple zones per region. (*Id.*, Ex. 9). There are, generally, multiple ways to split datasets. As alleged, Google's "approach is to partition database tables into contiguous ***key ranges*** called

[2] Kove alleges Spanner practices claim 1 of the '170 Patent (FAC ¶ 28); Colossus practices claim 15 of the '170 Patent (*id.* ¶ 39); and Spanner and Colossus together practice claim 17 of the '978 Patent (*id.* ¶¶ 61–77) and claim 10 of the '640 Patent (*id.* ¶¶ 82–98).

splits." (*Id.*, Ex. 15 at 1) (emphasis added); *see also id.*, Ex. 9; Ex. 17 at 5 ("Cloud Spanner distributes management of rows across multiple nodes by breaking up each table into several splits, ***using ranges*** of the lexicographically sorted primary key.")). This redundancy allows Spanner and Colossus to function even if a zone were to fail. (*Id.*, Ex. 9).

As seen below, once the data is split, a split ID is assigned to each key range.



In this example, we have a table with a simple integer primary key.

Split	KeyRange
0	[-∞,3)
1	[3,224)
2	[224,712)
3	[712,717)
4	[717,1265)
5	[1265,1724)
6	[1724,1997)
7	[1997,2456)
8	[2456,∞)

(*id.*, Ex. 15 at 3–4). Table splits can be referred to as "tablets." (*Id.*, Ex. 19 at 4). Both user tables/tablets (*i.e.*, the table/tablets that store a user's information) and user tablet location data can be split. (*Id.* at 4–5). As visualized below, user tablet location data is stored in metadata tablets. (*Id.* at 4). Sometimes, the queried metadata tablet does not know the user tablet's location information or the location information is incorrect. In those instances, the client "recursively moves up the tablet location hierarchy" without any sort of prompt or message from the tablet. (*Id.* at 5). In other words, the client will continue to query other metadata tablets within the same metadata table set for the requested user tablet location. (*Id.*).

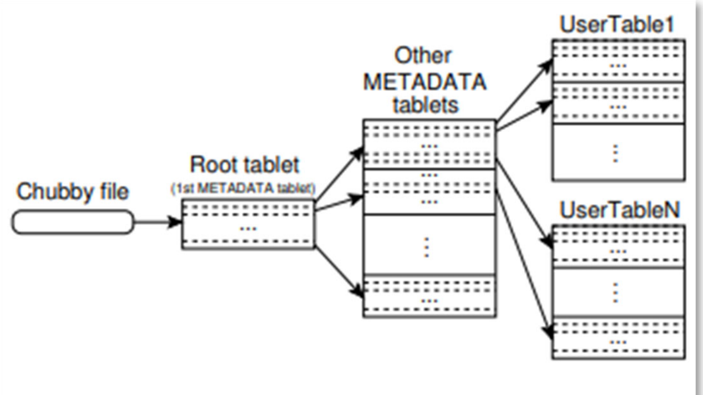


Figure 4: Tablet location hierarchy.

Using the above processes, the exhibits to the FAC show that Spanner and Colossus (i) store data and data's location, (ii) split data into multiple locations using a range-based approach, and (iii) retrieve data for a user using a hierarchy approach to locate the data.

II. ARGUMENT

A complaint must be dismissed if it fails to state a claim upon which relief may be granted. Fed. R. Civ. P. 12(b)(6). To survive a motion to dismiss, a complaint must contain sufficient factual matter to state a claim that is "plausible on its face." *Ashcroft v. Iqbal*, 556 U.S. 662, 678 (2009).

A. Kove Does Not Plausibly Allege Direct Infringement by Google.

To meet the plausibility standard, Kove must plead "factual content that allows the court to draw the reasonable inference that the defendant is liable for the misconduct alleged." *Iqbal*, 556 U.S. at 678. But "a patentee may subject its claims to early dismissal by pleading facts that are inconsistent with the requirements of its claims." *BOT M8 LLC v. Sony Corp. of Am.*, 4 F.4th 1342, 1346 (Fed. Cir. 2021) (citation omitted).

1. Because exhibits to the FAC show a hierarchy-based approach, Kove's allegations of a "redirect message" are facially implausible.

The allegations that Colossus infringed claim 15 of the '170 Patent and Spanner and Colossus infringed claim 1 of the '640 Patent are implausible for two reasons: (1) the FAC exhibits describe an altogether different method of finding information *without* any redirect message, and (2) even assuming a redirect message, Kove fails to plead facts that the alleged redirect message contains information used to identify the correct data location, as the claims require.

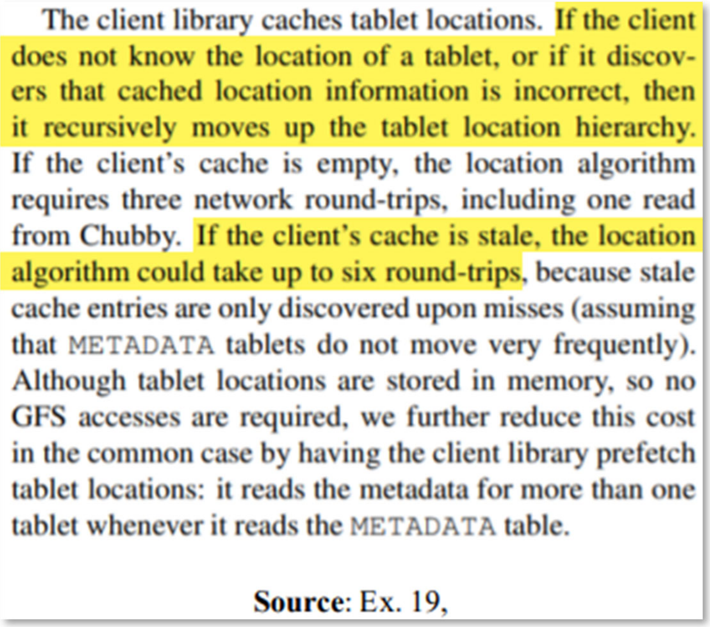
Claim 15 of the ’170 Patent requires “sending a redirect message;” similarly, Claim 10 of the ’640 Patent requires “transmitting a redirect message”:

Claim 15 of the ’170 Patent	Claim 10 of the ’640 Patent
A method of handling location queries in a network, the network comprising a plurality of location servers including data location information, the method comprising: correlating each one of a plurality of identifiers with at least one of a plurality of locations in the network, each one of the plurality of identifiers identifying a respective one of a plurality of data entities, wherein the data entities are stored in corresponding locations in the network; receiving a location query from a client at one of the plurality of location servers, the location query requesting location information identifying a location of a data entity included in the data entities; determining which of the plurality of location servers includes the location information; sending a location response message to the client in response to determining the one of the plurality of location servers includes the location information, the location response message comprising the location information; and **sending a redirect message to the client in response to determining the one of the plurality of location servers fails to include the location information, the redirect message identifying which of the plurality of**	A method for retrieving data location information for data stored in a distributed network, comprising the steps of: a) receiving at a first client a data query for retrieving data associated with an identification string, wherein the data is stored at a data repository in the distributed network and wherein a location string associated with the identification string of the data is stored in at least one of a plurality of data location servers; b) transmitting a data location request from the first client to a first data location server to retrieve the location string associated with the identification string in the data query, the data location request including the identification string; **c) if the first data location server does not possess the location string, transmitting a redirect message to the first client, the redirect message containing information for use by the first client to calculate a location of a second data location server, wherein the second data location server contains the location string;** d) calculating the location of the second data location server at the first client; and e) transmitting the data query from the first client to the second data location server. (*See* ’640 Patent, Claim 10) (emphasis added).

location servers includes the location information. (*See* ’170 Patent, Claim 15) (emphasis added).	

As shown above, both asserted claims require that if a queried location server or data location server does not have the required location information, then a “redirect message [is sent/transmitted] to [the/the first] client.” Furthermore, the claims require the redirect message to (1) identify a location server that includes the location information (’170 Patent), or (2) contain information the client can use to calculate the location of another server that contains the location information (’640 Patent). Kove alleges that the redirect message is the “cache miss response due to stale or incorrect information” sent to the client, (FAC ¶ 48), or, more abstractly, when “the client detects a miss.” (FAC ¶ 88). But Kove alleges that when the client receives a cache miss response, “the *location algorithm* [at the client] instructs the client to redirect, by moving up the tablet location hierarchy,” a process that “*may take six round-trips* . . . to determine the location of the correct tablet server.” (FAC ¶ 48 (citing Ex. 19) (emphases added)).

Kove erroneously cites the following screenshot of Exhibit 19 in support of its FAC:



The client library caches tablet locations. If the client does not know the location of a tablet, or if it discovers that cached location information is incorrect, then it recursively moves up the tablet location hierarchy. If the client’s cache is empty, the location algorithm requires three network round-trips, including one read from Chubby. If the client’s cache is stale, the location algorithm could take up to six round-trips, because stale cache entries are only discovered upon misses (assuming that METADATA tablets do not move very frequently). Although tablet locations are stored in memory, so no GFS accesses are required, we further reduce this cost in the common case by having the client library prefetch tablet locations: it reads the metadata for more than one tablet whenever it reads the METADATA table.

Source: Ex. 19,

(FAC ¶¶ 48, 90, and 91) (emphasis added). As illustrated above, Exhibit 19 demonstrates that the requesting *clients* (as opposed to being directed by a redirect message from responding *servers*) run a "location algorithm" to "recursively move up the tablet location hierarchy" to find the location. Thus, Kove conflates Exhibit 19's hierarchy-based approach with the claimed "redirect message" method of identifying data locations. These two approaches are incompatible (and, by extension, the infringement allegation is implausible). Again, Exhibit 19 describes a recursive process whereby a client uses a location algorithm to iteratively move up a tablet location hierarchy after every miss. In contrast, the claimed method requires sending a redirect message that *itself identifies* the data location to the client. Accordingly, the claimed "redirect message" is not only orthogonal to the hierarchy approach described in Exhibit 19 but also inconsistent with Exhibit 19's reference to "up to six round-trips." *Id.*, Ex. 19 at 5. Based on the plain language of the claims, if the Accused Products were using the claimed "redirect message," they would point the client to the correct location in response to the first request, making "up to six round-trips" unnecessary. Kove's allegation that the redirect message is a cache miss response contradicts Exhibit 19, so the Court does not need to credit the allegation. *Phillips v. Prudential Ins. Co. of Am.*, 714 F.3d 1017, 1020 (7th Cir. 2013) ("To the extent that an exhibit attached to or referenced by the complaint contradicts the complaint's allegations, the exhibit takes precedence.") (citation omitted).

Separately, Kove makes no factual allegations regarding the *contents* of the purported redirect message. The claims require the contents of the redirect message to include either the identity of "which of the plurality of location servers includes the location information" ('170 Patent, Claim 15), or "information for use by the first client to calculate a location of a second data location server . . . contain[ing] the location string" ('640 Patent, Claim 10). Put another way, Kove makes no allegation that the "cache miss response" (*i.e.*, the purported "redirect message") contains any of the information explicitly required of a redirect message because no factual allegation states that the "cache miss response" includes the identity of the correct location server (as required by '170 Patent, Claim 15) or information that a client (or even the location algorithm) can use to calculate a correct server location (as required by '640 Patent, Claim 10). This failure provides an independent basis to dismiss the infringement claims, even setting aside the fact that the recursive approach identified in the FAC is indeed different from the "redirect" technique claimed in the patents.

Because Exhibit 19 discloses a client that "recursively moves up the tablet location hierarchy"—which means the "location algorithm" at the client controls where it looks for the information according to a hierarchy, not a redirect message from a server—there is no reason to grant further leave to amend. It would be futile for Kove to try to re-plead given the known facts.

2. Claim 1 of the '170 Patent requires a hash function, but Kove cites exhibits showing the use of range-based identifiers, not hash-based identifiers.

The claim that Spanner infringes claim 1 of the '170 Patent is not plausible because Kove alleges infringement by use of hash-based identifiers whereas the cited exhibits show the use of range-based identifiers.

Claim 1 of the '170 Patent recites using a "hash function" to organize and retrieve data:

> A system for managing data stored in a distributed network, the system comprising:
>
> a data repository configured to store a data entity, wherein an identifier string identifies the data entity; and
>
> a data location server network comprising a plurality of data location servers, wherein data location information for a plurality of data entities is stored in the data location server network, at least one of the plurality of data location servers includes location information associated with the identifier string, each one of the plurality of data location servers comprises a processor and a portion of the data location information, **the portion of the data location information included in a corresponding one of the data location servers is based on a hash function used to organize the data location information across the plurality of data location servers**, and

> each one of the data location servers is configured to determine the at least one of the plurality of data location servers based on **the hash function applied to the identifier string.**[3]

(*See* '170 Patent, Claim 1) (emphasis added). Kove cites several exhibits showing Spanner using range-based identifiers ("key ranges"), not hash-based identifiers—a fact that even Kove admits. (*See* FAC ¶ 32 (citing Exs. 15 and 16) ("Google Spanner divides user data, or data entities, into chunks called splits, and these splits are stored at different Cloud Spanner data repository servers present at different locations **based on key ranges**.") (emphasis added); *id*. (Exhibit 15 figure showing a table split using key ranges); FAC ¶ 36 ("Spanner partitions tables into contiguous key ranges called splits and divide [sic] data among servers by key ranges.")).

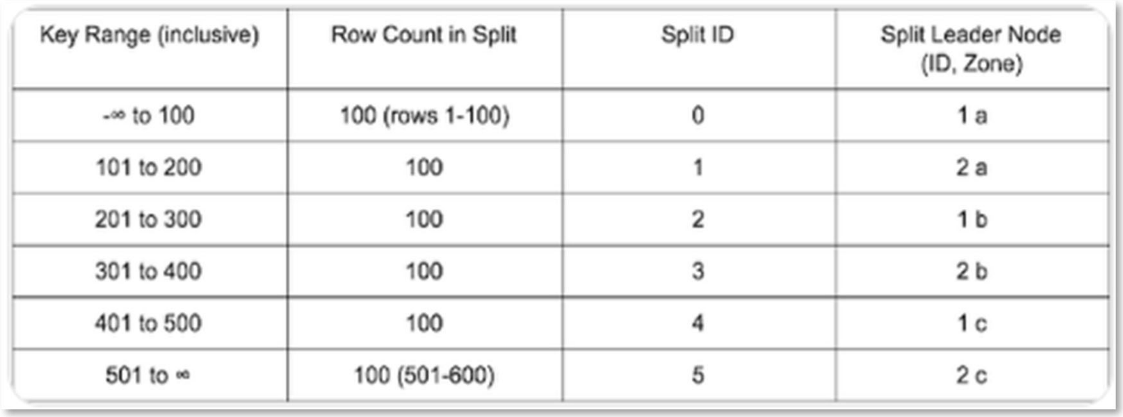
Kove's related allegations simply parrot the claim language and note that hash-based identifiers are referenced within Exhibits 15-18 as an option that *users* can implement. (FAC ¶¶ 33, 34, 36–38 (citing Exs. 15–18). Notably, however, allegations that Google induced or contributed to Google *users'* alleged infringement have been withdrawn in the FAC, and what remains does not support any inference of direct infringement. The exhibits control, *Phillips v. Prudential*, 714 F.3d at 1020, and they show that Spanner used range-based identifiers, not hash-based identifiers.

The option described in Exhibits 15-18 to use hash-based identifiers is only one among various techniques that users can implement. And the reason that hotspots may occur is attributable to the very fact that Spanner divides data by (unhashed) key ranges. (FAC ¶ 33 and 36) (both citing Ex. 16, which explains that "Cloud Spanner divides data among servers by key ranges, which

[3] Here, too, claim construction is unnecessary. The phrase "based on a hash function used to organize the data location information across the plurality of data location servers . . . based on the hash function applied to the identifier string" was construed as "the portion of the data location information included in a corresponding one of the data location servers is based on a hash function that maps identifier strings to one or more of the data location servers, and each one of the data location servers is configured to determine the at least one of the plurality of data location servers based on the hash function applied to the identifier string." *Kove*, Dkt. 484 at 24–30, 35.

means your inserts will be directed at a single server, creating a hotspot.”);[4] (FAC ¶ 37) (citing Ex. 15 which states that an application developer “can put a hash value in their primary key” to avoid “the performance cost of potential row-range hotspots”). Put another way, the accused Spanner product used range-based identifiers that could lead to hotspots with certain types of data, which hotspots users could choose to avoid by implementing hash-based identifiers.

Specifically, Kove’s citation to Exhibit 17 shows no plausible infringement. Exhibit 17 teaches a limited context wherein ***users can utilize hashing*** when they decide to implement an “integer sequence as a key” (also relevant to avoiding hotspots caused by Spanner’s range-based approach), and goes so far as to say that this approach is “not recommended.” (FAC, Ex. 17 at 8–10). Again, Exhibit 17 makes clear that Spanner uses ranges, not hashes. For example, under the header “Table splits and key choice,” Exhibit 17 states that “Cloud Spanner distributes management of rows across multiple nodes by breaking up each table into several splits, *using ranges of the lexicographically sorted primary key*.” (*Id.*, Ex. 17 at 3–5) (emphasis added). Further, Exhibit 17 depicts the use of “Key Range[s],” using this example at pages 5-6:



Key Range (inclusive)	Row Count in Split	Split ID	Split Leader Node (ID, Zone)
-∞ to 100	100 (rows 1-100)	0	1 a
101 to 200	100	1	2 a
201 to 300	100	2	1 b
301 to 400	100	3	2 b
401 to 500	100	4	1 c
501 to ∞	100 (501-600)	5	2 c

[4] Key ranges are mentioned again in Exhibit 16—“A split holds a range of contiguous rows”—when describing how Spanner “splits” data (FAC, Ex. 16 at 3; *id.* ¶ 34). Exhibit 16 is replete with the examples that primary keys are sorted “contiguously” (*i.e.*, next in a sequence), which is a defining characteristic of range-based identifiers. (*Id.*, Ex. 16 at 1, 7, and 9). Exhibit 16 therefore demonstrates that Spanner uses “key ranges,” and not hashes.

Similarly, Exhibit 15 outlines the salient differences between range- and hash-based approaches, explaining the range-based approach is the one used by Spanner, but that users can write their applications to instead implement a hash-based approach:

> Another example is [Spanner's] data layout, where ***we use range sharding***. Applications that do range scans can use this layout to get high performance. But some applications don't need range scans; and ***if those applications don't want to worry about the performance cost of potential row-range hotspots, they can put a hash value in their primary key, effectively causing Spanner to use hash sharding***. In general, Spanner has been designed to offer powerful tools to application builders, but to ***give those builders a high degree of control*** over any tradeoffs between powerful functionality and performance.

(*id.*, Ex. 15 at 10) (emphases added). Kove cites Exhibit 15 in support of its allegation that Spanner utilizes hash-based identifiers to avoid hotspots. (FAC ¶¶ 36 and 37). But, as with Exhibit 17, Exhibit 15 says that Google's Spanner uses range-based sharding but users can implement hash-based sharding; there are no allegations that Google itself does hash-based sharding, and any claims of indirect infringement by users have been withdrawn. *Id.*, Ex. 15 at 10.

In sum, once allegations that contradict the FAC's exhibits are set aside, all that is left is an allegation that Spanner uses range-based identifiers. Because the Asserted Claim requires hash-based sharding, Kove has pleaded no plausible direct infringement claim against Google.

3. Claim 17 of the '978 Patent requires a "transfer protocol," but Kove provides tautological allegations and fails to allege that the "transfer protocol" transports both identifier and location information.

The claim that Spanner and Colossus infringed claim 17 of the '978 Patent is not plausible because Kove failed to (1) identify a "transfer protocol" and instead attempted to cure its originally deficient claim by adding a meaningless allegation, and (2) allege that the "transfer protocol" transports *both* identifier and location information, as required by the claim.

Claim 17 of the '978 Patent recites providing a "transfer protocol" to transport identifier and location information, and storing the location information in the format of the protocol:

> A method of scaling at least one of capacity and transaction rate capability in a location server in a system having a plurality of location servers **for storing and retrieving location information**, wherein each of the plurality of location servers stores unique set of location information of an aggregate set of location information, the method comprising:
>
> **providing a transfer protocol configured to transport identifier and location information, the location information specifying the location of information related to the identifier;**
>
> **storing location information formatted according to the transfer protocol at a first location server**;
>
> receiving an identifier and a location relevant to the identifier at the first location server;
>
> storing the received location in a location store at the first data location server, the location store comprising a plurality of identifiers, each identifier associated with at least one location, wherein the received location is associated with the received identifier in the location store; and
>
> transferring a portion of the identifiers and associated locations to a second data location server when a performance criterion of the first location server reaches a predetermined performance limit.

(*See* '978 Patent, Claim 17) (emphasis added). Thus, claim 17 requires that the transfer protocol be "configured to transport identifier and location information" and define the format of stored location information. *Id*. Kove's FAC insufficiently alleges facts related to either requirement.

i. Kove alleges insufficient facts identifying a "transfer protocol" and what little it does allege is tautological.

Instead of pleading facts identifying the claimed "transfer protocol" in response to Google's prior motion, Kove's FAC just added a tautological allegation that the steps of storing and retrieving certain location data must be a transfer protocol "because all the requisite network communications occur[.]" (FAC ¶ 62). But, as well-known transfer protocols like FTP (file transfer protocol), HTTP (hyper-text transfer protocol), and TCP/IP (the Internet protocol suite comprised of a transmission control protocol and Internet protocol) make clear, a transfer protocol defines the procedures by which computers communicate data between one another, not just the fact that data is transferred. Consistent with that understanding, Claim 17 explicitly requires more than a communication occurring: the claimed "transfer protocol" must be configured in a certain way, and location information must be stored in a format defined by that transfer protocol.

Aside from alleging that a transfer protocol must exist because communication occurs, Kove alleges a "transfer protocol" only by reference to other steps of Claim 17—*i.e.*, "storage" and "retrieval" of location information—rather than as a separate, independent limitation. Kove's amendment therefore insufficiently pleads facts alleging a transfer protocol separate and apart from other claim limitations. *BOT M8 LLC v. Sony Corp. of Am.*, 4 F.4th 1342, 1352 (Fed. Cir. 2021) ("[A] plausible claim must do more than merely allege entitlement to relief; it must support the grounds for that entitlement with sufficient factual content.").

Finally, in reference to those other claim limitations, such as "storage" of location information, Kove alleges that stored location information is transferred; however, even then, Kove's allegations are inconsistent and confusing. For example, Kove alleges that Colossus alone provides the transfer protocol "on a request to recover a tablet and its location information stored in a Metadata table" and that the transfer protocol is "[t]he steps of storage and retrieval of location information stored in BigTable" that begins by "a tablet server (node) reading its metadata from the Metadata table." (FAC ¶ 62). Here, too, Kove merely points to the storage and retrieval of information rather than anything about how data is transferred or what provides the transfer protocol. (*Id.* (citing Exs. 8 and 19)). Kove then seemingly changes horses in midstream to allege that "retriev[ing] data from [a] table" is the "exemplary transfer protocol" and that this is completed by "BigTable us[ing] Rowkeys." (FAC ¶ 66). Finally, in regards to the "formatted according to the transfer protocol" limitation, Kove claims that "BigTable uses Chubby services to keep track of tablet servers" and "BigTable stores tablet location information in tablet servers" (FAC ¶ 64 (citing Ex. 19, at 4)), but then states "BigTable uses the SSTable format to store data," apparently signifying that the SSTable format is the the "format[] according to the transfer protocol." (FAC ¶ 65). Again, Kove's transfer protocol allegations either (1) consist of references to steps of storage and retrieval as outlined by *other* claim limitations, or (2) fail to provide a set of consistent factual allegations that separately and clearly identify a transfer protocol.

These random, inconsistent allegations do not add up to a coherent claim, much less put Google on notice of how it supposedly infringed. More is required as a matter of pleading under *Iqbal* and *Twombly*. When the asserted claim requires "providing a transfer protocol configured to transport identifier and location information" and "storing location information formatted according to the transfer protocol," it does not suffice as a matter of pleading to just say the Accused Products store and retrieve data. Kove must plead facts sufficient to support its allegations about the claimed "transfer protocol."

ii. Kove fails to allege that the "transfer protocol" transports both identifier and location information.

Here, too, there may be a simpler basis for the Court to dismiss. Claim 17 requires that the provided "transfer protocol" be "configured to transport identifier and location information." Kove perfunctorily alleges that the "Accused Products provide a transfer protocol configured to transport identifier and location information." (FAC ¶ 61). But that conclusory allegation, which just mimics the claim language, can be disregarded. *Iqbal*, 556 U.S. 662, 678 ("The allegations are conclusory and not entitled to be assumed true."). Other than that, Kove's FAC omits any allegation that the transfer protocol is configured to transport or transports "identifier information." Instead, Kove alleges only that "[t]he steps of storage and retrieval of ***location information*** stored in BigTable is a transfer protocol." (FAC ¶ 62) (emphasis added).

III. CONCLUSION

For the foregoing reasons, Google respectfully requests that the Court dismiss Kove's FAC for failure to state a claim upon which relief can be granted.

Dated: December 18, 2023

Respectfully submitted,

/s/ Robert W. Unikel
Robert W. Unikel (Bar No. 6216974)
Douglas L. Sawyer (Bar No. 6275849)
Mark T. Smith (Bar No. 6315040)
John A. Cotiguala (Bar No. 6311056)
Grayson S. Cornwell (Bar No. 6338845)
Summer C. Stevens (Bar No. 6342946)
PAUL HASTINGS LLP
71 S. Wacker Drive, Suite 4500
Chicago, Illinois 60606
Tel: (312) 499-6000
Fax: (312) 499-6100
robertunikel@paulhastings.com
dougsawyer@paulhastings.com
marksmith@paulhastings.com
johncotiguala@paulhastings.com
graysoncornwell@paulhastings.com
summerstevens@paulhastings.com

Counsel for Defendant,
GOOGLE LLC

CERTIFICATE OF SERVICE

I hereby certify that a true and correct copy of the foregoing document was filed electronically in compliance with Fed. R. Civ. P. 5(a) on December 18, 2023. As of this date, all counsel of record had consented to electronic service and are being served with a copy of this document through the Court's CM/ECF system under Fed. R. Civ. P. 5(b) and (c).

/s/ *Robert W. Unikel*
Robert W. Unikel